(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 852 419 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
08.07.1998 Bulletin 1998/28

(51) Int. Cl.^{6}: H02J 13/00

(21) Application number: 97121370.7

(22) Date of filing: 04.12.1997

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 04.12.1996 IL 11975396

(71) Applicant:
Powercom Control Systems Ltd.
Givat Shmuel 51905 (IL)

(72) Inventor: Liberman, Isidor
Givat Shmuel 51905 (IL)

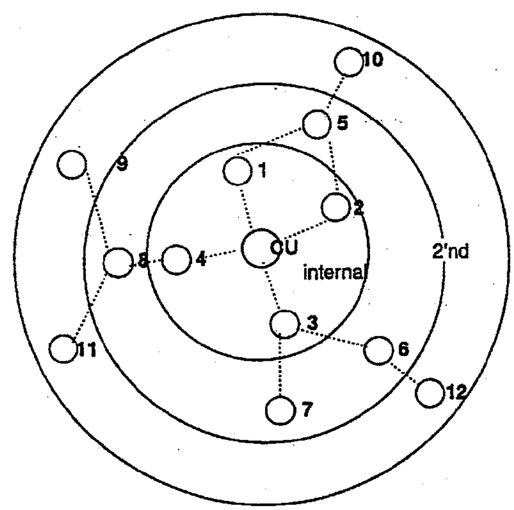
(74) Representative:
Bohnenberger, Johannes, Dr. et al
Meissner, Bolte & Partner
Postfach 86 06 24
81633 München (DE)

(54) Electric power supply management system

(57) A communication protocol is provided for centralized power line networks, which allows for automatic network mapping and adaptive routing, for automatic log-on of remote nodes and also for optimization of communication routes and control of the traffic volume.

A central unit (CU) polls all the known remote nodes (RN) and a polled RN replies with a message. A new remote node (RN), which hears a polling command from the CU, transmits a Burst-Log-On Message direct to the CU. A new RN, which hears a reply of polled RN, but not the polling message itself, transmits a Burst Log-On Message, using the replying RN as a relay. A route may include several relays between the new RN and the CU and all of them are registered in the Log On message itself. When the CU receives the message from the new RN, it performs a log on of the RN, stores its route, and polls it in the next polling cycle. This process is continuous, enabling the addition of new nodes. RN with no communication behaves as a new one and creates a new route.

Figure 2 - Mapping Idea Example



EP 0 852 419 A2

Automatic Log-On And Mapping

Basic Definitions

The protocol defines the behavior of 2 system segments: the **Central Unit (CU)** and the **Remote Node (RN).** **The CU operation is built up from 2 operational modes:**

- **Polling mode** - when the CU sends polling commands to all known nodes;
- **Burst mode** - when it is waiting for the nodes log-on messages. The CU starts to work in the Polling mode. If no node is registered in the data base it starts to poll an imaginary RN.

RN also operates in 2 modes:

- **Known mode** - RN receives polling messages from the CU and replays according to the received command.
- **Lost/New mode** - when RN is newly installed in the network or is lost, it did not get a polling message for 2 interrogation cycles or more. In this case it sends a Burst log-on message via a received route.

MAPPING PROCESS

General Description

Mapping means to find routes for all nodes. The system functions as follows:

1. The mapping process is constantly performed by the CU in both modes of operation: polling and burst. The difference is in the expected quantity of new nodes. There is no special mode for mapping.
2. At the beginning all nodes are in the **New/Lost** mode and there are no known nodes in the CU data base. The CU starts the polling cycle by sending a polling message to an imaginary node, with a reserved node ID. (There is no influence on the process if such an ID really exists.)
3. The polling message contains a time for the bursts start and a number of available time slots.
4. Each node, which receives the message, stores the CU as its neighbor, enters the **Known mode** and waits for a time to transmit a Log-on Burst Message. The time for burst is computed by each **New/Lost** node as follows: Time For Burst Start:, as received from the message, plus some function of node ID, in the range of the number of slots.
5. The CU enters the burst mode and waits. Upon receiving a reply message, it records the replying node in the data base as one that has a direct connection with the CU.
6. When the burst mode is finished, the CU computes new parameters for the burst mode: start time and a number of slots and enters the polling mode. Now it performs its first real interrogation cycle, transmitting polling messages to all the nodes in its data base. At this stage all the nodes are with direct connection, without relays in the route.
7. Each **New/Lost** node that receives the message stores again the CU as its neighbor. Each **New/Lost** node which did not get this message, but is done receive the polling nodes reply stores the last heard node as its neighbor.
8. All the nodes wait for the time of burst and transmit the Log-on Burst Messages to their neighbors. If the neighbor is the CU the procedure is as before. If it is another node it acts as relay and sends in turn the received message to its neighbor. At this time the relay maybe only a node which already did respond to the CU polling message and has a direct connection with the CU.
9. Now the procedure continues as follows:

   - CU stores each new node received by it in the data base together with the route (one relay in the case above).
   - In the polling mode, all the nodes are interrogated by sending to them polling messages via the stored routes.
   - Each **New/Lost** node, which receives some node reply to the CU, or it's transmission as relay, registers it as its neighbor.
   - When the burst time comes, it transmits its log-on message to its neighbor.
   - Each node that receives a message, transmits it again to its neighbor, until the message arrives the CU.

MAPPING PROCESS EXAMPLE

The mapping process will be clarified in the following example: As can be seen in Figure 2 it is a system which is divided into 3 circuits. Nodes in the internal circuit hear the CU directly. Nodes in the 2'nd circuit hear some nodes from

node and the CU compute statistics for the known routes. The statistic takes into account the number of retries from the previous cycles, when their weights are smaller and smaller as the time progresses.

CREATING DYNAMIC ROUTES BY THE BEST NEIGHBOR METHOD

From remote node

In a static communicational environment, the approach of the best route may be good enough and the network after some time will achieve a state of the best route for each node. In the Electrical Power Line networks, the communication probability varies due to the electrical loads and line noises, and a relay that was good some times, maybe bad when required.

For example, the following situation as described in figure 3.

Suppose that the Node 4 had received the following route at the time T1: node 3, node 1, CU, as the best the route. However, at time T2, when the Node 4 wants to create a connection with CU, there is no communication between 3 and 1, but a good connection between 3 and 2. Node 4 has very little chance to know this, because it has no connection at all with 2 & 1. Node 3 in order has a good chance to know it because it should receive all the Node 1 and Node 2 transmissions. So it is much better for node 4 to send the message to Node 3 and not to specify all the route to CU. Node 3 in order, transmits the message to the next relay, 2 or 1, according to its up-to-date situation.

CREATING DYNAMIC ROUTES IN CU

The problem of creating dynamic routes, as described above for a remote node, exists also in the CU and is even a little bit more complicated. The CU is a single point in the network and each remote node knows how to transmit the message to it via its best neighbor. However, when the CU transmits the message to a remote node, it have to specify full route as the nodes (See figure 4).

An additional problem is that the CU gets the routes from received messages at different periods of the day and in general the best night routes, (according to the minimal time criteria), are not good during the day.

If the CU keeps the route to the nodes as it received it, it is impossible to connect node 4 via the route received at night.

The CU acts as follows:

- For each node it keeps only the best neighbors, as received from the message;
- For each neighbor it keeps a statistic for relevant time periods (6 periods of 4 hours are enough);
- The information is arranged as follows:

Node	Neighbors	period 1	period 2	...	period N
ND_1	ND_{11}	p_{111}	p_{211}		p_{N11}
	ND_{12}	p_{112}	p_{212}		p_{N12}
	......				
	ND_{1m}	p_{11m}	p_{21m}		p_{N1m}
ND_2	ND_{21}	p_{121}	p_{221}		p_{N21}
	ND_{22}	p_{122}	p_{222}		p_{N22}
	ND_{2m}	p_{12m}	p_{22m}		p_{N2m}
....	..	...	..	...	...
..	..	...	..	...	...
..	..	...	..	...	...
ND_n	ND_{n1}	p_{1n1}	p_{2n1}		p_{Nn1}
	ND_{n2}	p_{1n2}	p_{2n2}		p_{Nn2}
	......				

when it is in the installation phase and many nodes start to send their log-on messages, this protocol does not work and the stations jam each other.

Polling is the basic communication method. The CU transmits the polling commands to all the known nodes in the system. Nodes respond only to the polling command, so there is no mutual interference between them. In large systems, with a poor time synchronization, this is the only method that enables safe periodical data transferring. This method however does not enable new nodes to join the system and the data collection from nodes with a low communication probability is hard.

In the **Slotted Burst Method** the nodes may send messages upon event in specified time slots. The CU computes the start and the number of slots of the burst period, according to the time required for the polling cycle and the expected number of new nodes. After this, CU transmits these parameters in each polling message. They are repeated by each relay and target node and received by New/Lost nodes. Nodes that do not receive any message shall not send any burst message as the probability that somebody will hear them is very low.

Generally, we want to perform the burst period only after finishing the polling cycle for all known nodes. However, the time between bursts should be limited to some value that represents the communication stability. If the time is exceed there is a high probability, that for a large part of New/Lost nodes the roots they did receive during the polling cycle, will be no more available. In the PLC case the time should be no longer than half an hour.

The Number of Slots is initialized to 4 and it is always the smallest number of available slots. Each slot is 10 seconds wide, which in average enables 20 not interfered retries of one node.

Each New/Lost node, after receiving the burst period parameters, computes its time for log-on transmission as follows:

T= Bursts Start Time + (ND_{id})modulo. Number Of Slots

where NDid is the identification number of node.

During the burst period, the CU waits for the bursts messages and does not send polling commands at all. When the burst period is finished, the CU computes the new number of slots as the number of New/Lost nodes * 3.

The reasoning for this is as follows:

When a large number of nodes are added at once, a rapid enlarges of the Burst Time is required. This may happen for example when a new CU is installed in the network or a branch of electric cable is moved from one transformer to another.

Increasing the burst time period, enables more new nodes to join the system without interference, but on the other hand this reduces the time for polling, which in turn decreases the chances of nodes to receive proper routes. When there are many Burst Log-on messages, we assume that there are also many new nodes and we may assume that because of collisions we do not receive a part of them. Increasing the Burst Time as described, seems to be a good solution as the time increases rapidly as long as the number of new nodes increases and is reduced rapidly as the number decreases.

HELPING MESSAGES

The methods described above may be insufficient when an On Line connection with nodes is required because the communication conditions may change dramatically from the last received route.

The idea is that nodes may volunteer to help the received message to reach their goal (the target node, relay or CU). This method generates a high number of messages transmitted in parallel and should only be used for messages with high priority.

The messages may be sent forward, from the CU to node, or backward, from a node to the CU. The treatment for each case is different.

HELPING BACKWARD MESSAGES

This case is very simple because each known node in the network knows the way to the CU through the list of neighbors it keeps. If the node receives a message addressed to the CU it checks if there is an Ack to this message. If after a number of retries no Ack was received the nodes will transmit the message to their neighbors.

HELPING FORWARD MESSAGES

The case when the message is transmitted from the CU is more complicated. This is because in general the nodes have a limited memory and they have no possibility to keep all the communication connections. Additionally, during its work the CU is all the time transmitting polling messages permanently giving information about its communication position relative to the receiving nodes. This is not the case with the nodes as they are generally silent until receiving a pole command or some other event.

5) the central unit enters the burst mode and waits to receive a reply message, it records the replying node (RN) in the data base as one that has a direct connection with the central unit,

6) when the burst mode is finished, the central unit computes new parameters for the burst mode including start time and a number of slots and re-enters the polling mode, upon which it performs its first real interrogation cycle, transmitting polling messages to all the nodes in its data base, at this stage all the nodes are with direct connection, without relays in the route,

7) the New/Lost mode of each node (RN) that receives the polling message stores again the central unit as its neighbour, while each New/Lost node which did not get this message but receives the polling reply from another node (RN) stores the last heard node as its neighbour,

8) all the nodes wait for the time of burst and transmit the Log-on Burst Messages to their neighbours, wherein if the neighbour is the central unit the procedure is as before, but if it is another node it acts as a relay and sends in turn the received message to its neighbour,

9) the above procedure continues, wherein:

- central unit stores each new node received by it in the data base together with the route (one or more relay nodes),
- in the polling mode of the central unit (CU), all the nodes are interrogated by sending to them polling messages via the stored routes,
- each node in the New/Lost mode receives some node reply to the central unit, or its transmission as relay, registers it as its neighbour,
- when the burst time comes, it transmits its log-on message to its neighbour, and
- each node that receives a message, transmits it again to its neighbour, unit the message arrives at the central unit.

2. A system as claimed in Claim 1, where optimization of routes is achieved by the following steps:

a) central unit sends polling messages to all the nodes in its data base, each message containing the number of retries performed for this message,

b) central unit initializes this field to 0 and increments it with each new transmission, until receiving an acknowledge (Ack) from the first relay on the route,

c) each relay on the route also increments this field with each re-transmission as does the target node, when replying to the central unit command,

d) each message contains a unique ID, built up from two fields: the Cycle Number and the Message Sequence Number in the cycle,

e) each node processes all the messages in the network, even though it should not respond to them, as a relay or a target, while each node maintains a list of routes, each route keeping the Message ID from which it has achieved the route and the number of retries specified in the message the first time received,

f) the routes are sorted by a node according to the number of retries, where the route with the smallest number of retries it receives is the best for it,

g) the next time the node responds to the central unit command, or it sends a burst message, it will do this via the new route, and the central unit will save the last received route for it next polling command,

h) to prevent a route, which accidentally succeeds to be the best one, the node and the central unit compute statistics for the known routes, the statistics accounting for the number of retries from the previous cycles, with

Figure 1 - typical PLC network

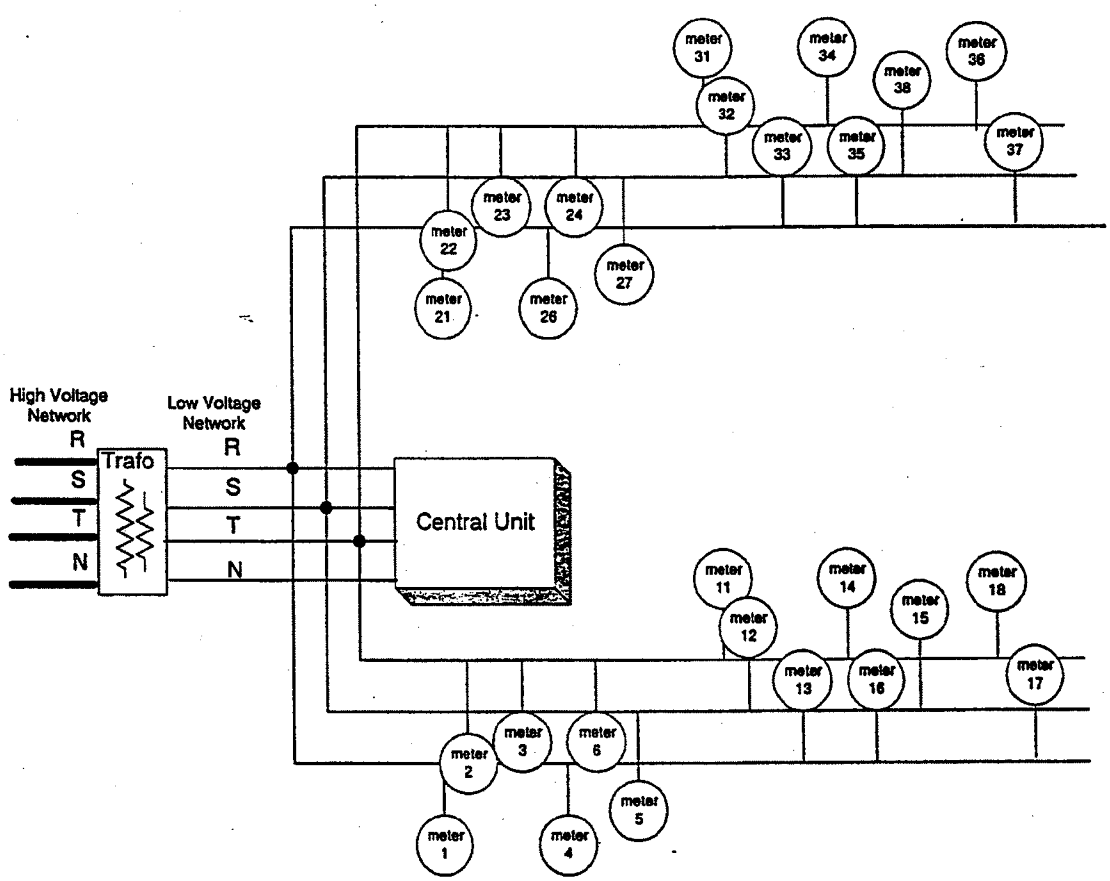


Figure 3 - Choosing the Best Neighbor

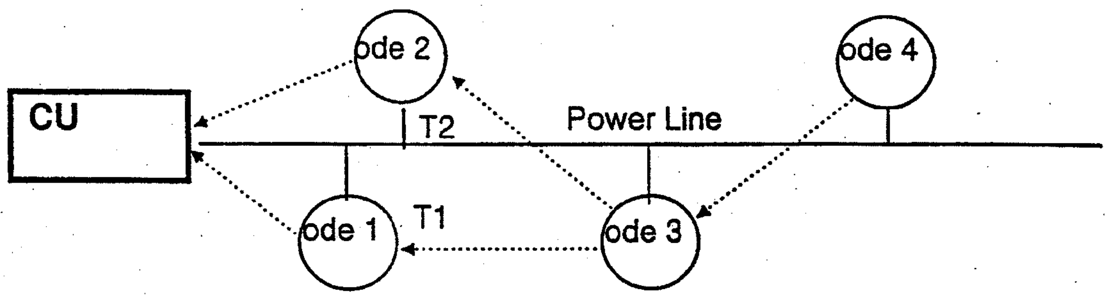
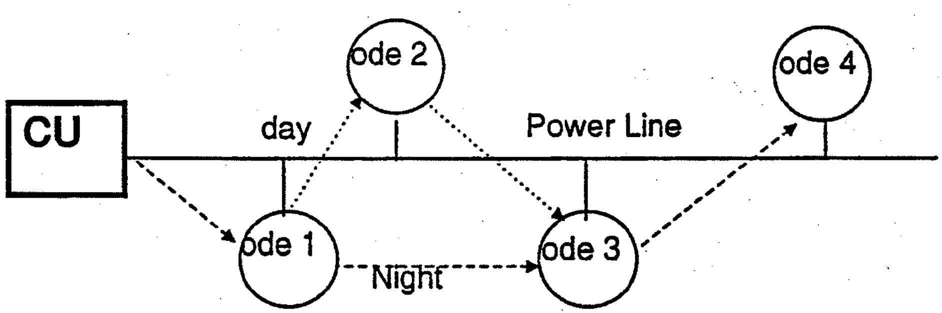


Figure 4 - Creating Routes in CU



(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 852 419 A3

(12) **EUROPEAN PATENT APPLICATION**

(88) Date of publication A3:
09.12.1998 Bulletin 1998/50

(43) Date of publication A2:
08.07.1998 Bulletin 1998/28

(21) Application number: **97121370.7**

(22) Date of filing: **04.12.1997**

(51) Int. Cl.[6]: **H02J 13/00**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **04.12.1996 IL 11975396**

(71) Applicant:
Powercom Control Systems Ltd.
Givat Shmuel 51905 (IL)

(72) Inventor: **Liberman, Isidor**
Jerusalem (IL)

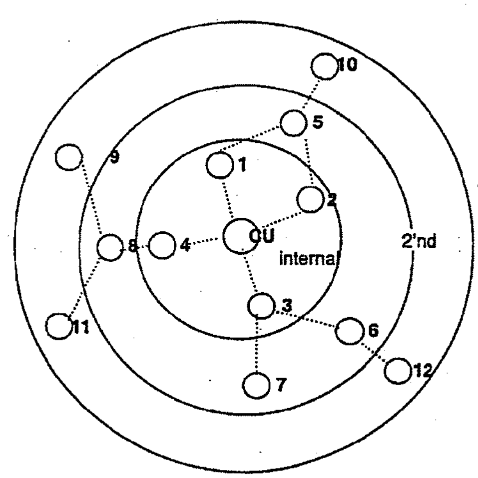
(74) Representative:
Bohnenberger, Johannes, Dr. et al
Meissner, Bolte & Partner
Postfach 86 06 24
81633 München (DE)

(54) **Electric power supply management system**

(57) A communication protocol is provided for centralized power line networks, which allows for automatic network mapping and adaptive routing, for automatic log-on of remote nodes and also for optimization of communication routes and control of the traffic volume.

A central unit (CU) polls all the known remote nodes (RN) and a polled RN replies with a message. A new remote node (RN), which hears a polling command from the CU, transmits a Burst-Log-On Message direct to the CU. A new RN, which hears a reply of polled RN, but not the polling message itself, transmits a Burst Log-On Message, using the replying RN as a relay. A route may include several relays between the new RN and the CU and all of them are registered in the Log On message itself. When the CU receives the message from the new RN, it performs a log on of the RN, stores its route, and polls it in the next polling cycle. This process is continuous, enabling the addition of new nodes. RN with no communication behaves as a new one and creates a new route.

Figure 2 - Mapping Idea Example



EP 0 852 419 A3